

NETWORK DISCOVERY

BACKGROUND

[0001] A known solution for discovering a network, for example a managed network, collected data from all managed nodes in the network and then processed all of the collected data. When a managed network contains a sufficiently large number of managed nodes, bulk discovery of the nodes can increase the memory resident size of the central discovery process beyond resources of the system hosting the central discovery process.

SUMMARY

[0002] A method for discovering a network comprising network devices, includes dividing the network into zones of network devices, and in a zone of the network, identifying devices in the zone that have SNMP (Simple Network Management Protocol) access, collecting data from the identified devices, and stitching the collected data into a topology of the network. A machine readable medium can include software or a computer program or programs for causing a computing device to perform the exemplary method. An exemplary system for discovering a network organized into zones of network devices includes means for identifying devices in a zone of the network that have SNMP access, collecting data from those devices in the zone identified as having SNMP access, and stitching the collected data into a topology of the network, and means for transferring data to and from the means for identifying, collecting and stitching.

BRIEF DESCRIPTION OF THE DRAWINGS

[0003] The accompanying drawings provide visual representations which will be used to more fully describe the representative embodiments disclosed

herein and can be used by those skilled in the art to better understand them and their inherent advantages. In these drawings, like reference numerals identify corresponding elements and:

[0004] Figure 1 illustrates a flow diagram in accordance with an exemplary method.

[0005] Figure 2 illustrates a functional diagram in accordance with an exemplary method.

[0006] Figure 3 illustrates a functional diagram in accordance with an exemplary method.

[0007]

DETAILED DESCRIPTION

[0008] Figure 1 illustrates an exemplary process for discovering a network, for example a managed network including network devices having SNMP (Simple Network Management Protocol) access. In a first block 100, the network is divided into zones of network devices. In a next block 102, in a zone of the network, devices in the zone that have SNMP (Simple Network Management Protocol) access are identified. In a next block 104, data are collected from the identified devices, *i.e.*, those devices that were identified as having SNMP access in the zone. In a next block 106, the collected data are stitched into a topology of the network. In a next block 108, a determination is made whether all of the zones in the network have been processed. If no, then control returns to block 102, where the process can be started anew with respect to a different zone of the network. If yes, then control proceeds to block 110, where the topology is downloaded into a database. From block 110, control proceeds to block 112 where the process ends.

[0009] The subprocess in block 108 can be performed, for example, by incrementing a number representing a zone identification and processing the corresponding zone, until all zones have been processed. The subprocess in block

104 can be performed using agents, for example agents to which nodes in an identified zone have been dispatched or identified. The agents can collect data from network devices at the dispatched nodes using SNMP, for those nodes having SNMP access. The subprocess of block 100 can be performed, for example, manually by an Administrator (for example via the seed file 210 shown in Figure 2), and/or by processes variously described in the following copending applications, which are hereby incorporated by reference: a U.S. non-provisional patent application entitled "METHOD AND SYSTEM FOR DETERMINING A NETWORK MANAGEMENT SCALABILITY THRESHOLD OF A NETWORK MANAGER WITH RESPECT TO A NETWORK", filed in the U.S. Patent and Trademark Office on 23 September 2003 and bearing Attorney Docket identification number "200311141-1" (032842-141), inventors Gabriel Brandon Wechter, Eric A. Pulsipher, and Max Carl Knees; and a U.S. non-provisional patent application entitled "METHOD AND SYSTEM FOR MANAGING A NETWORK OF NODES", filed in the U.S. Patent and Trademark Office on 23 September 2003 and bearing Attorney Docket identification number "200311662" (032842-148), inventors Gabriel Brandon Wechter, Eric A. Pulsipher, and Max Carl Knees.

[0010] Figure 2 illustrates discovery of a first zone in a managed network, in accordance with an exemplary embodiment. As shown in Figure 2, a managed environment or network 202 is connected to or administered to by a software application or module 204 such as Hewlett Packard's Network Node Manager (NNM), which is in turn connected to a software module or application 206 also named "ovet_bridge". The ovet_bridge 206 receives a list or batch of managed nodes of the network 202 from the NNM 204, and publishes it to a module or data element (such as a file) 212, also named "NNM hosts". The data element 212 can include, for example, an IP (Internet Protocol) address and node name on each line. A software module or application or element 214 reads data from the module

204, and inserts nodes into an ovet_disco finders.returns table in a database 216. A network administrator 208 can also provide a seed file or seed data 210 describing nodes in the network 202, to the module 214. An Ovet discovery or "ovet_disco" process 270 includes multiple modules or processes, for example the modules or processes 216, 218, 220, 222, 232, 234, 238, 240 and 242, as shown in Figure 2 and described herein.

[0011] A FinderRetProcessing stitcher 218 is invoked on each insertion to the finder.returns table of the database 216, and moves or inserts the node to a finders.processing portion of the database 216. A FinderProcToDetailsDesp stitcher 220 is invoked on each insertion to the finders.processing portion of the database 216, and moves nodes to a Details.dispatch portion of a database 222. A details agent 224 receives nodes from the Details.dispatch portion of the database 222, and performs a set of SNMP queries (for example, to nodes in the network 202 having SNMP access) for values such as sysDescr and sysObjectID, and inserts values received in response to the queries into a Details.returns portion of the database 222. For example, the details agent 224 can collect or indicate SNMP accessible nodes. In an exemplary embodiment this can increase efficiency of subsequent processing, because non-SNMP nodes can be ignored. When the details agent 224 completes its survey of all nodes, a ZoneProcessing stitcher 232 is invoked.

[0012] The ZoneProcessing stitcher 232 uses information from the Details.returns portion of the database 222 to compute a list of zones in the network 202, and dispatches valid nodes (for example, nodes having sysObjectIDs) in the first zone to active agent(s) 236. This is performed using an Agents.dispatch portion or table of a database 234, which is accessed by or transfers information to the agents 236. Agent dispatch table insertions into the Agents.dispatch portion or table of the database 234 can be automatically or transparently rejected if their corresponding sysObjectIDs are not supported by

one of the agents 236. The ZoneProcessing stitcher 232 can set a zone count, a zone ID (identification), a "first zone" flag, and an "all zones done" flag in a Disco.zones portion or table of a database 242 before returning. The agent(s) 236 use the data from the Agents.dispatch portion or table of the database 234 to collect data, and the agent(s) 236 populate the collected data into an Agents.returns table or portion of the database 234. The Agents.returns table or portion of the database 234 can act as a cache for this information. An AgentsToWorkEntities stitcher 238 can be activated upon every insertion into the Agent.returns table of the database 234, and can forward data from the insertion to a workingEntities.entityByName portion or table of a database 240. The database 240 can act as a cache for the forwarded data.

[0013] The agent(s) 236 can be separate entities or binaries from the ovet_disco process, and can be specific to or can be configured to interact effectively with, the various switches, nodes, and protocols found on or supported within the network 202.

[0014] Figure 3 illustrates an exemplary process with respect to subsequent processing of zones, after the first zone. As shown in Figure 3, a FinalPhase stitcher 302 is invoked, for example after the agent(s) 236 have completed data collection for a zone. The FinalPhase stitcher connects to multiple stitchers, including an IPv6FinalPhase stitcher 304 and a ScratchTopologyToStorage stitcher 306. The ScratchTopologyToStorage stitcher 306 processes the discovery topology data and downloads a scratchTopology.entityByName table to a master entityByName table, which is managed by an OQL (Object Query Language) adapter. The master entityByName table can act as a cache for this information. The ScratchTopologyToStorage stitcher 306 also transfers data to an IPv6TopologyToModel stitcher 308. The ScratchTopologyToStorage stitcher 306 then invokes a ZoneComplete stitcher 310.

[0015] The ZoneComplete stitcher 310, which can be a text stitcher, clears the Agents.dispatch and Agents.returns tables of the database 234, refreshes a topology and layer database (which can be connected, for example, to the ScratchTopologyToStorage stitcher 306), clears SNMP and DNS helper caches, and signals ovet_disco that this zone cycle has been completed. If all zones are not yet done, as indicated for example by the "all zones done" flag in the database 242, then ovet_disco can invoke the ZoneProcessing stitcher 232 to continue the processes shown in Figure 3 and the processes shown in Figure 2 with respect to elements 232, 234, 236, 238, 240 and 242 for a next zone. The ZoneProcessing stitcher 232 can increment a zone ID value to indicate a next zone for processing, and then perform the procedures described above.

[0016] An exemplary process of identifying devices in the network that have SNMP access can include a first module (e.g., 206) receiving a list of managed nodes in the network and publishing the list of managed nodes to a first file (e.g., 212), a second module (e.g., 214) reading the first file and inserting data from the file into a returns portion of a first database (e.g., 216), invoking a third module (e.g., 218) upon each insertion of data from the first file into the returns portion of the first database, which inserts data from the returns portion of the first database into a processing portion of the first database, invoking a fourth module (e.g., 220) upon each insertion of data into the processing portion of the first database, the fourth module identifying nodes corresponding to the inserted data to a dispatch portion of a second database (e.g., 222), and a details agent (e.g., 224) obtaining node identifications from the dispatch portion of the second database, performing queries to the nodes corresponding to the node identifications, and inserting information received in response to the queries into a returns portion of the second database.

[0017] An exemplary process of collecting data from the identified devices includes invoking a fifth module (e.g., 232) which accesses the returns portion of

the second database, computes a list of the zones, and dispatches valid nodes in the first zone or the zone currently being processed to active agents (e.g., 236) via a dispatch portion of a third database (e.g., 234), and the agents collecting data from the valid nodes and returning the collected data to a returns portion of the third database.

[0018] An exemplary process can also include invoking a sixth module (e.g., 302, 306), which causes the collected data in the returns portion of the third database to be processed into discovery topology data of the network and then downloaded, and invoking a seventh module (e.g., 310), which clears the dispatch and returns portions of the third database and refreshes topology and layer databases and signals that topological analysis with respect to the zone has been completed.

[0019] Using and stitching together zones of a network, allows much larger environments to be discovered and managed than situations where an entire network is discovered all at once. A cache management mechanism can also be provided to allow accurate connectivity and to allow logical relationships to be deduced and represented across discoveries of multiple zones. In accordance with exemplary embodiments, a user can specify through a graphical user interface (GUI), for example the GUI 404 shown in Figure 4, how the network or environment is to be divided into zones, and/or a zone configuration can be obtained via other mechanisms, for example as described in the copending applications incorporated herein by reference.

[0020] The zone configuration can be used or consumed by the discovery mechanism(s) described herein. When a discovery is initiated, each device in the network can be tested against the zone configuration and then assigned one or more zone identifiers. Each zone can be discovered in turn. A node or device belonging to multiple zones would be a multi-zone device, and can function to link or organize zones. For example, information about multi-zone devices can be used

to keep connectivity information about the network current, and can allow switches to be properly grouped into VLAN(s) (Virtual Local Area Network). Discovery algorithms can thus automatically track the zones to which the various entities in the network belong, and can piece together the zones into a topology representing the entire environment. For example, at the end of discovery of each zone, the resulting "scratch topology" can be examined by an algorithm that performs the following:

[0021]

1 For each object in the scratch topology:

1.1 If the object is a "node entity"

1.1.1 If the node is part of multiple zones

1.1.1.1 Remember the name of this node in a ZoneCache (temporary) database.

1.1.2 If the node has not already been discovered in a previous zone

1.1.2.1 Send the node record to a persistent topology database.

1.2 If the object is an "interface entity" (e.g., a component of a node entity)

1.2.1 If the interface is part of a node in the ZoneCache database AND the interface is directly connected to another node

1.2.1.1 Add the connection information for this interface to the ZoneCache database.

1.2.1.2 Update the existing record in the persistent topology database.

2 At the conclusion of the final zone, write container objects (e.g., VLANs, HSRP (Hot Standby Router Protocol) groups) associated with multiple zones to the persistent topology database and prepare for consumption by downstream applications (e.g., graphical views, path analysis, and event correlation algorithm(s)).

[0022] The ZoneCache database can facilitate updating the connections associated with an already-downloaded node, for example a multi-zone node.

[0023] Those skilled in the art will also appreciate that the elements and methods or processes described herein can be implemented using a microprocessor, computer, or any other computing device, and can be implemented in hardware and/or software, in a single physical location or in distributed fashion among various locations or host computing platforms. The agents can be implemented in hardware and/or software at any desired or appropriate location. For example, Figure 4 also shows a computer 402 connected to the GUI 404 and to the network 202, as well as software 408 and NNM software 406 operating on the computer 402. The software 408 can include modules for performing the processes shown in Figures 1-3.

[0024] The processes and mechanisms described herein, for example with respect to Figures 1-3, can be implemented individually or collectively on the computer 402, which can be implemented using a microprocessor, computer, or any other computing device, and can be implemented in a single physical location or in distributed fashion among various locations or host computing platforms. Agents such as the agents 224, 236, and stitchers such as the stitchers 218, 220, 232, 238, 302, 304, 306, 308, and 310 can be implemented in hardware and/or software at any desired or appropriate location. The computer 402 can also include data transfer or input/output devices of various kinds, including a keyboard, and one or more connections to computing resources (data and/or processing) external to the computer 402. The various databases shown in Figures 2-3 can be implemented as memory within the computer 402 or accessible to the computer 402, for example as Random Access Memory (RAM), one or more hard drives located locally and/or in distributed fashion through a network or internet, or any data storage medium or mechanism.

[0025] An exemplary system for discovering a network organized into zones of network devices, includes means (for example, the computer 402 with software 406, 408) for identifying devices in a zone of the network that have

SNMP (Simple Network Management Protocol) access, collecting data from those devices in the zone identified as having SNMP access, and stitching the collected data into a topology of the network, and means for transferring data to and from the means for identifying, collecting and stitching (for example, the GUI 404 or another other data transfer mechanism, interface, etc.). The means for caching data can be implemented via RAM within the computer 402 or accessible to the computer 402, and/or via one or more hard drives located locally and/or in distributed fashion through a network or internet, or via any data storage medium or mechanism.

[0026] It will also be appreciated by those skilled in the art that the present invention can be embodied in other specific forms without departing from the spirit or essential characteristics thereof, and that the invention is not limited to the specific embodiments described herein. The presently disclosed embodiments are therefore considered in all respects to be illustrative and not restrictive. The scope of the invention is indicated by the appended claims rather than the foregoing description, and all changes that come within the meaning and range and equivalents thereof are intended to be embraced therein.